

III. OS 설정

1. /etc/hosts 파일에 설치할 서버의 IP와 호스트명을 추가합니다.(root)

1. /etc/hosts 파일을 열어서 서버의 ip 주소와 호스트명 추가

```
vi /etc/hosts
```

```
[root@SON ~]# vi /etc/hosts
[root@SON ~]#
[root@SON ~]# cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1        localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.56.160 SON
```

2. hostname 을 설정합니다.

```
hostnamectl set-hostname SON
```

```
[root@SON ~]# hostnamectl set-hostname SON
```

3. hostname 및 현재 설치된 리눅스 버전을 함께 확인합니다.

```
# hostnamectl status
Static hostname: SON
    Icon name: computer-vm
    Chassis: vm
    Machine ID: 12cdc87e047342729df9d285b2d131fe
    Boot ID: d822a922bc614ceda19e695bc675c87d
    Virtualization: oracle
Operating System: Red Hat Enterprise Linux 8.8 (Ootpa)
    CPE OS Name: cpe:/o:redhat:enterprise_linux:8::baseos
    Kernel: Linux 4.18.0-477.10.1.el8_8.x86_64
    Architecture: x86-64
[root@SON ~]#
```

```
[root@SON ~]# hostnamectl status
  Static hostname: SON
        Icon name: computer-vm
        Chassis: vm
        Machine ID: 12cdc87e047342729df9d285b2d131fe
        Boot ID: d822a922bc614ceda19e695bc675c87d
  Virtualization: oracle
  Operating System: Red Hat Enterprise Linux 8.8 (Ootpa)
        CPE OS Name: cpe:/o:redhat:enterprise_linux:8::baseos
        Kernel: Linux 4.18.0-477.10.1.el8_8.x86_64
  Architecture: x86-64
```

2. Local repo 설정(root)

1. 현재 생성되어 있는 repo 파일 확인

```
[root@SON ~]# cd /etc/yum.repos.d/
[root@SON yum.repos.d]# ls
[root@SON yum.repos.d]#
```

```
[root@SON ~]# cd /etc/yum.repos.d/
[root@SON yum.repos.d]# ls
[root@SON yum.repos.d]#
```

- 파일이 있다면? : 외부 통신을 사용하지 않을 것이기 때문에 기존에 존재하는 repo 파일 삭제(`rm *.repo`)
- 파일이 없다면? : repository 등록을 위해 iso 파일을 리눅스에 마운트

2. iso 파일을 리눅스에 마운트

모바텀이나 putty를 이용해서 업로드해도 되고, VM에서 iso파일을 마운트해도 된다.

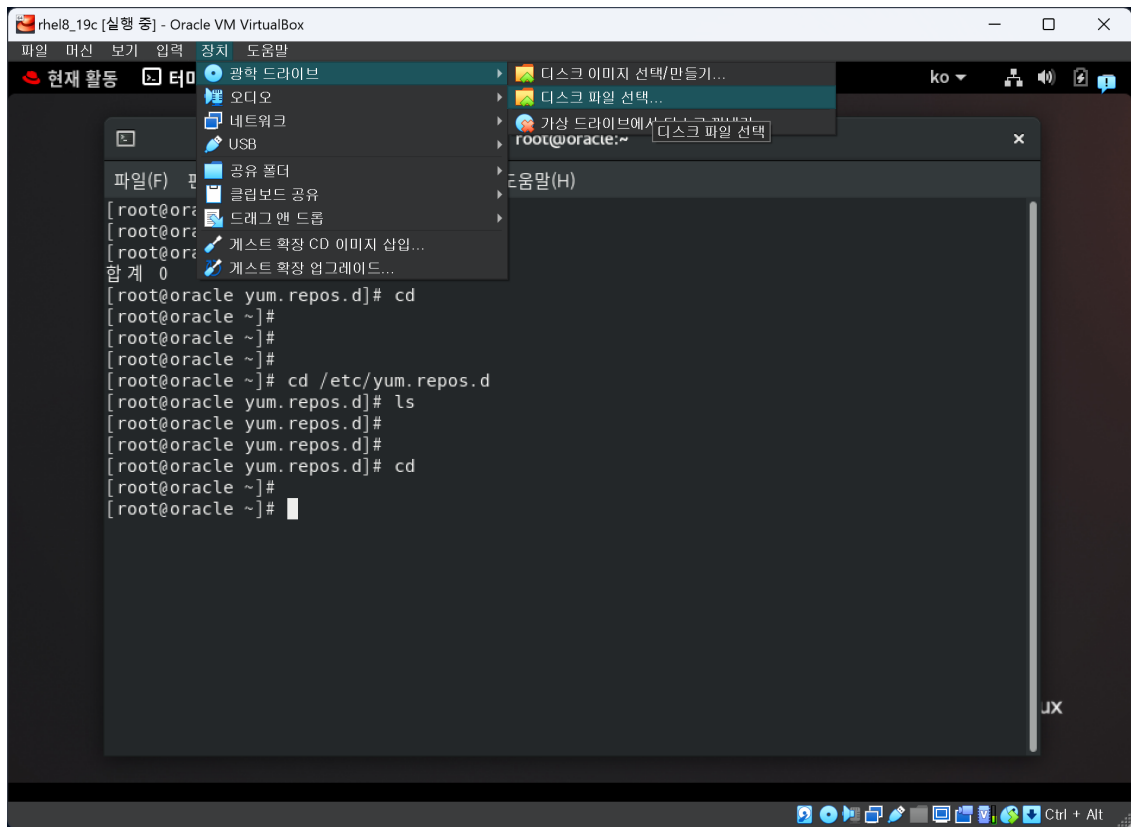
VirtualBox의 경우 리눅스에서 장치-광학 드라이브-디스크 파일 선택 을 통해 리눅스 설치 iso파일을 올린다.



*** iso 파일을 올리는 이유**

: iso 파일의 Packages 디렉토리에는 서버를 운영하기 위한 기본적인 rpm파일들이 들어있다.

▼ 과정 캡처



임의의 경로에 마운트 된 것을 확인합니다.

```
df -h
```

```
[root@S0N ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        3.9G   0  3.9G   0% /dev
tmpfs           3.9G   0  3.9G   0% /dev/shm
tmpfs           3.9G  9.3M  3.9G   1% /run
tmpfs           3.9G   0  3.9G   0% /sys/fs/cgroup
/dev/sda3       72G   6.1G  66G   9% /
/dev/sda1       507M  226M  281M  45% /boot
tmpfs          796M   20K  796M   1% /run/user/0
/dev/sr0        12G   12G   0 100% /run/media/root/RHEL-8-8-0-BaseOS-x86_64
```

3. iso 파일을 마운트할 /root/repo 디렉토리를 생성합니다.

위의 iso이미지 파일을 마운트 디렉토리 /root/repo를 생성합니다.
repo에 구성될 패키지 파일을 둘 디렉토리가 필요하기 때문입니다.

```
# iso 파일을 마운트할 디렉토리 생성
mkdir /root/repo

# 잘 생성 되었는지 확인
ls -ld /root/repo
```

```
[root@SON ~]# mkdir /root/repo
[root@SON ~]#
[root@SON ~]# ls -ld /root/repo
drwxr-xr-x. 2 root root 6  6월 20 15:57 /root/repo
```

4. iso 파일을 생성한 디렉토리에 마운트 합니다.

```
# iso 파일을 /root/repo에 마운트
mount -o loop /dev/sr0 /root/repo

# 임시 디렉토리에 잘 마운트 되었는지 확인
ll /root/repo
```

- -o : 옵션
- loop : iso 파일을 마운트 할 때 사용하는 옵션

따라서 /dev/sr0에 있는 iso 파일을 /root/repo에 마운트

```
[root@SON ~]# mount -o loop /dev/sr0 /root/repo
[root@SON ~]#
[root@SON ~]# ll /root/repo
합 계 48
dr-xr-xr-x. 4 root root 2048 4월 11 2023 AppStream
dr-xr-xr-x. 4 root root 2048 4월 11 2023 BaseOS
dr-xr-xr-x. 3 root root 2048 4월 11 2023 EFI
-r--r--r--. 1 root root 8154 4월 11 2023 EULA
-r--r--r--. 1 root root 18092 4월 11 2023 GPL
-r--r--r--. 1 root root 1669 4월 11 2023 RPM-GPG-KEY-redhat-beta
-r--r--r--. 1 root root 5135 4월 11 2023 RPM-GPG-KEY-redhat-release
-r--r--r--. 1 root root 1796 4월 11 2023 TRANS.TBL
-r--r--r--. 1 root root 1455 4월 11 2023 extra_files.json
dr-xr-xr-x. 3 root root 2048 4월 11 2023 images
dr-xr-xr-x. 2 root root 2048 4월 11 2023 isolinux
-r--r--r--. 1 root root 103 4월 11 2023 media.repo
```

5. /root/repo 디렉토리 패키지를 이용해 local repo 파일을 생성합니다.

file:/// : 이 접두사는 파일 URI(Uniform Resource Identifier)를 나타냅니다.

이는 로컬 파일 시스템에 있는 파일을 참조하는 데 사용됩니다.

#예시

```
[ 저장소 섹션 이름 ]
name= repo 이름
baseurl= 저장소의 로컬 경로
enabled=저장소 활성화 상태(비활성화 0, 활성화 1)
gpgcheck= GPG 키 검증 상태(비활성화 0, 활성화 1)
gpgkey=패키지 무결성을 검증하기 위해 사용할 GPG 키의 경로 지정
```

- gpgcheck=1으로 했을 때

```
# vi /etc/yum.repos.d/local.repo

[local-BaseOS]
name=BaseOS
baseurl=file:///root/repo/BaseOS
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

[local-AppStream]
name=AppStream
baseurl=file:///root/repo/AppStream
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-rehat-release
```

3. 필수 패키지 설치 및 확인(root)

1. repo를 불러오고 필수 패키지를 설치합니다.

```
dnf install -y bc
dnf install -y binutils
dnf install -y elfutils-libelf
dnf install -y elfutils-libelf-devel
dnf install -y fontconfig-devel
dnf install -y glibc
dnf install -y glibc-devel
dnf install -y ksh
dnf install -y libaio
dnf install -y libaio-devel
dnf install -y libXrender
dnf install -y libX11
dnf install -y libXau
dnf install -y libXi
dnf install -y libXtst
dnf install -y libgcc
dnf install -y libnsl
dnf install -y librdmacm
dnf install -y libstdc++
dnf install -y libstdc++-devel
dnf install -y libxcb
dnf install -y libibverbs
dnf install -y make
dnf install -y policycoreutils
dnf install -y policycoreutils-python-utils
dnf install -y smartmontools
dnf install -y sysstat
```

▼ MOS 에 기재된 필수 패키지

Oracle Linux 8 / RHEL 8

OS Version	Patches/Packages	Kernel settings
<p>Oracle Linux 8.1 with the Unbreakable Enterprise Kernel 6: 5.4.17-2011.0.7.el8uek.x86_64 or later</p> <p>Oracle Linux 8.1 with the Red Hat Compatible kernel: 4.18.0-147.el8.x86_64 or later</p> <p>Red Hat Enterprise Linux 8.1: 4.18.0-147.el8.x86_64 or later</p> <p>Reference :</p> <p>Note 2738613.1</p>	<p>bc binutils compat-openssl10 elfutils-libelf elfutils-libelf-devel glibc glibc-devel ksh libaio libaio-devel libXrender libX11 libXau libXi libXtst libgcc libnsl librdmacm libstdc++ libstdc++-devel libxcb libibverbs make smartmontools sysstat</p>	<p>kernel.shmall=Greater than or equal to the value of shmmmax, in pages /proc/sys/kernel/shmall kernel.shmmmax=Half the size of physical memory in bytes /proc/sys/kernel/shmmmax kernel.sem = 250 32000 100 128 fs.file-max = 6815744 fs.aio-max-nr = 1048576</p> <p>Note: This value limits concurrent outstanding requests and should be set to avoid I/O subsystem failures.</p> <p>net.ipv4.ip_local_port_range=9000 65500 net.core.rmem_default=262144 net.core.rmem_max=4194304 net.core.wmem_default=262144 net.core.wmem_max=1048576</p> <p>/etc/security/limits.conf</p> <p>oracle soft nproc 2047 oracle hard nproc 16384 oracle soft nofile 1024 oracle hard nofile 65536 oracle soft stack 10240 oracle hard stack 10240</p>

2. 필수 패키지 설치 확인

```
rpm -qa | grep bc
rpm -qa | grep binutils
rpm -qa | grep elfutils-libelf
rpm -qa | grep elfutils-libelf-devel
rpm -qa | grep fontconfig-devel
rpm -qa | grep glibc
rpm -qa | grep glibc-devel
rpm -qa | grep ksh
rpm -qa | grep libaio
rpm -qa | grep libaio-devel
rpm -qa | grep libXrender
rpm -qa | grep libX11
rpm -qa | grep libXau
rpm -qa | grep libXi
rpm -qa | grep libXtst
rpm -qa | grep libgcc
rpm -qa | grep libnsl
rpm -qa | grep librdmacm
rpm -qa | grep libstdc++
rpm -qa | grep libstdc++-devel
rpm -qa | grep libxcb
rpm -qa | grep libibverbs
rpm -qa | grep make
rpm -qa | grep policycoreutils
```

```
rpm -qa | grep policycoreutils-python-utils
rpm -qa | grep smartmontools
rpm -qa | grep sysstat
```

4. OS 계정 및 GROUP 확인 (root)

```
vi /etc/passwd
vi /etc/group
```

없을 경우 생성하고, 각 유저는 dba 그룹에 속하도록 구성

- oracle 유저 : DB HOME 유저
- grid 유저 : GRID HOME 유저
- dba 그룹 : 최고 권한을 가진 그룹이자 설치 되는 것들을 관리하는 그룹

```
# dba 그룹만 생성
groupadd dba

# oracle 사용자 생성 및 dba 그룹에 추가
useradd -g dba -G dba oracle

# grid 사용자 생성 및 dba 그룹에 추가
useradd -g dba -G dba grid

# oracle 유저 passwd 생성 (oracle)
passwd oracle

# grid 유저 passwd 생성 (grid)
passwd grid

# 구성 후 확인
vi /etc/passwd
vi /etc/group

id oracle
id grid
```

```

[root@SON ~]# vi /etc/passwd
[root@SON ~]#
[root@SON ~]# vi /etc/group
[root@SON ~]#
[root@SON ~]# groupadd dba
[root@SON ~]#
[root@SON ~]# useradd -g dba -G dba oracle
[root@SON ~]#
[root@SON ~]# useradd -g dba -G dba grid
[root@SON ~]#
[root@SON ~]# passwd oracle
oracle 사용자의 비밀번호 변경 중
새 암호 :
잘못된 암호 : 암호는 8 개의 문자 보다 짧습니다
새 암호 재입력 :
passwd: 모든 인증 토큰이 성공적으로 업데이트 되었습니다.
[root@SON ~]#
[root@SON ~]# passwd grid
grid 사용자의 비밀번호 변경 중
새 암호 :
잘못된 암호 : 암호는 8 개의 문자 보다 짧습니다
새 암호 재입력 :
passwd: 모든 인증 토큰이 성공적으로 업데이트 되었습니다.
[root@SON ~]#
[root@SON ~]# id oracle
uid=1001(oracle) gid=1001(dba) groups=1001(dba)
[root@SON ~]#
[root@SON ~]# id grid
uid=1002(grid) gid=1001(dba) groups=1001(dba)

```

5. THP 비활성화(root)

Transparent Huge Page는 RHEL에서 기본적으로 사용 설정이 되어있으나, Oracle 성능 이슈로 인해 비활성화 권장

(근거 : <https://docs.oracle.com/en/database/oracle/oracle-database/19/ldb/ldbi/disabling-transparent-hugepages.html#GUID-02E9147D-D565-4AF8-B12A-8E6E9F74BEEA>)

```

echo never > /sys/kernel/mm/transparent_hugepage/enabled
cat /sys/kernel/mm/transparent_hugepage/enabled

```

위 명령어는 현재 부팅된 세션에서 THP를 비활성화합니다. 따라서 시스템을 재부팅하면 설정이 원래대로 돌아갑니다.

```

[root@SON ~]# echo never > /sys/kernel/mm/transparent_hugepage/enabled
[root@SON ~]# cat /sys/kernel/mm/transparent_hugepage/enabled
always madvise [never]
[root@SON ~]#

```

영구적으로 비활성화 하는 방법은 RAC 설치 과정에서 보여드리도록 하겠습니다.

6. 커널 파라미터 세팅(root)

(공식 문서 권장 값 : <https://docs.oracle.com/en/database/oracle/oracle-database/19/ldb/ldbi/changing-kernel-parameter-values.html>)

```

# 커널 값 세팅
vi /etc/sysctl.conf

# 커널 값 적용
sysctl -p

```



```
fs.aio-max-nr = 1048576
fs.file-max = 6815744
kernel.shmall = 2097152
kernel.shmmax = 4294967295
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
```

```
[root@SON ~]# sysctl -p
fs.aio-max-nr = 1048576
fs.file-max = 6815744
kernel.shmall = 2097152
kernel.shmmax = 4294967295
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
```

7. Resource limits 제한(root)

운영 체제를 사용하여 파일의 리소스 제한

- 참고 공식 문서 : <https://docs.oracle.com/en/database/oracle/oracle-database/19/ladbi/checking-resource-limits-for-oracle-software-installation-users.html>

Use the following ranges as guidelines for resource allocation to Oracle software installation owners for standard installations:

Table 5-1 Oracle Software Installation Owner Resource Limit Recommended Ranges for Standard Installations

Resource Shell Limit	Resource	Soft Limit	Hard Limit
Open file descriptors	nofile	at least 1024	at least 65536
Number of processes available to a single user	nproc	at least 2047	at least 16384
Size of the stack segment of the process	stack	at least 10240 KB	at least 10240 KB, and at most 32768 KB
Maximum locked memory limit	memlock	unlimited	unlimited

```
vi /etc/security/limits.conf
```

⇒ 계정 별 혹은 그룹 별 리소스 제한을 걸어주는 파일

```
oracle soft nofile 1024
oracle hard nofile 65536
oracle soft nproc 2047
oracle hard nproc 16384
oracle soft stack 10240
oracle hard stack 32768
oracle hard memlock unlimited
oracle soft memlock unlimited
```

```
grid soft nofile 1024
grid hard nofile 65536
grid soft nproc 2047
grid hard nproc 16384
grid soft stack 10240
grid hard stack 32768
grid hard memlock unlimited
grid soft memlock unlimited
```

파일 디스크립터 수 제한 확인

```
$ ulimit -Sn
```

```
1024
```

```
$ ulimit -Hn
```

```
65536
```

프로세스 수 제한 확인

```
$ ulimit -Su
```

```
2047
```

```
$ ulimit -Hu
```

```
16384
```

스택 크기 제한 확인

```
$ ulimit -Ss
```

```
10240
```

```
$ ulimit -Hs
```

```
32768
```

8. 디렉토리 생성 및 권한 설정(root)

1. 디렉토리 생성

```
# Oracle DB HOME 및 디렉토리 생성
mkdir -p /u01/app/oracle
mkdir -p /u01/app/oracle/product/dbhome_1

# Oracle Inventory directory 생성
mkdir -p /u01/app/oraInventory
```

```
# Oracle Grid Infrastructure 기본 홈 및 디렉토리 생성
mkdir -p /u01/app/grid
mkdir -p /u01/app/grid/product/grid
```

2. 디렉토리 소유권 및 권한 설정

```
chown -R grid:dba /u01/app/grid
chown -R oracle:dba /u01/app/oracle
chown -R grid:dba /u01/app/oraInventory

chmod -R 755 /u01
```

```
[root@SON ~]# mkdir -p /u01/app/oracle
[root@SON ~]# mkdir -p /u01/app/oracle/product/dbhome_1
[root@SON ~]#
[root@SON ~]# mkdir -p /u01/app/oraInventory
[root@SON ~]#
[root@SON ~]# mkdir -p /u01/app/grid
[root@SON ~]# mkdir -p /u01/app/grid/product/grid
[root@SON ~]#
[root@SON ~]# chown -R grid:dba /u01/app/grid
[root@SON ~]# chown -R oracle:dba /u01/app/oracle
[root@SON ~]# chown -R grid:dba /u01/app/oraInventory
[root@SON ~]#
[root@SON ~]# chmod -R 755 /u01
[root@SON ~]#
```

9. 환경 변수 설정(각 유저 모두 설정)

1. Oracle 유저의 .bash_profile 설정

```
# su - oracle
$ vi .bash_profile

-----
PATH=$PATH

umask 022
export PATH
export PS1='[${ORACLE_SID}]@\u:\w]$ '

export ORACLE_SID=SONNY
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=$ORACLE_BASE/product/dbhome_1
export PATH=$ORACLE_HOME/bin:$PATH:/usr/bin:$ORACLE_HOME/OPatch
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/usr/lib:$ORACLE_HOME/network/lib
export CV_ASSUME_DISTID=RHEL7
export LANG=C

-----

$ source .bash_profile
```

2. GRID 유저의 .bash_profile 설정

```
# su - grid
$ vi .bash_profile
-----

PATH=$PATH

umask 022
export PATH
export PS1='[${ORACLE_SID}]@u:\w]$ '

export ORACLE_SID=+ASM
export ORACLE_BASE=/u01/app/grid
export ORACLE_HOME=$ORACLE_BASE/product/grid
export PATH=$ORACLE_HOME/bin:$PATH:/usr/bin:$ORACLE_HOME/OPatch
export CV_ASSUME_DISTID=RHEL7
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/usr/lib:$ORACLE_HOME/network/lib
export LANG=C

-----

$ source .bash_profile
```

10. selinux 변경(root)

1. sellinux 설정 값 확인

```
[root@SON ~]# sestatus
SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:       /etc/selinux
Loaded policy name:            targeted
Current mode:                  enforcing
Mode from config file:        enforcing
Policy MLS status:             enabled
Policy deny_unknown status:    allowed
Memory protection checking:    actual (secure)
Max kernel policy version:     33
```

2. 프로세스 동작 차단 가능성을 방지하기 위해서 변경

```
[root@son ~]# vi /etc/selinux/config
```

```
# /etc/selinux/config 파일 내용 예시

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
```

```
SELINUX=disabled
# SELINUXTYPE= can take one of these three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

3. 변경 사항 확인

```
[root@SON ~]# sestatus
SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:      /etc/selinux
Loaded policy name:           targeted
Current mode:                 permissive
Mode from config file:       disabled
Policy MLS status:           enabled
Policy deny_unknown status:   allowed
Memory protection checking:   actual (secure)
Max kernel policy version:    33
```

```
[root@SON ~]# sestatus
SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:      /etc/selinux
Loaded policy name:           targeted
Current mode:                 enforcing
Mode from config file:       disabled
Policy MLS status:           enabled
Policy deny_unknown status:   allowed
Memory protection checking:   actual (secure)
Max kernel policy version:    33
```

11. 방화벽 차단(root)

1. 방화벽 차단

```
systemctl stop firewalld
systemctl disable firewalld
```

```
[root@SON ~]# systemctl stop firewalld
[root@SON ~]# systemctl disable firewalld
Removed /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
```

2. 방화벽 상태 확인

```
# systemctl status firewalld
```

```
[root@SON ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
   Active: inactive (dead) since Thu 2024-06-20 17:29:45 KST; 23s ago
     Docs: man:firewalld(1)
   Main PID: 722 (code=exited, status=0/SUCCESS)

6월 20 15:44:27 SON systemd[1]: Starting firewalld - dynamic firewall daemon...
6월 20 15:44:30 SON systemd[1]: Started firewalld - dynamic firewall daemon.
6월 20 15:44:32 SON firewalld[722]: WARNING: AllowZoneDrifting is enabled. This is considered an insecure configuration option. It will be removed i>
6월 20 17:29:45 SON systemd[1]: Stopping firewalld - dynamic firewall daemon...
6월 20 17:29:45 SON systemd[1]: firewalld.service: Succeeded.
6월 20 17:29:45 SON systemd[1]: Stopped firewalld - dynamic firewall daemon.
```

```
[root@SON ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
   Active: inactive (dead) since Thu 2024-06-20 17:29:45 KST; 23s ago
     Docs: man:firewalld(1)
   Main PID: 722 (code=exited, status=0/SUCCESS)

6월 20 15:44:27 SON systemd[1]: Starting firewalld - dynamic firewall daemon...
6월 20 15:44:30 SON systemd[1]: Started firewalld - dynamic firewall daemon.
6월 20 15:44:32 SON firewalld[722]: WARNING: AllowZoneDrifting is enabled. This is considered an insecure configuration option. It will be removed i>
6월 20 17:29:45 SON systemd[1]: Stopping firewalld - dynamic firewall daemon...
6월 20 17:29:45 SON systemd[1]: firewalld.service: Succeeded.
6월 20 17:29:45 SON systemd[1]: Stopped firewalld - dynamic firewall daemon.
```